

Introdução á programação

João Gonçalves

João Gonçalves

- Interactive Developer / Trainer / Consultant
 - Adobe Certified Expert & Instructor
 - Applied Mathematics Degree
 - Master Multimedia Communication
-
- <http://joaogoncalves.net>
 - edu@joaogoncalves.net
 - [@joaopapin](#)



FLAG

You?



Partilha

git....

Lógica

Sequência Lógica

Algoritmo

Simbologia

Diagramas

Constantes

Variáveis

Lógica

Lógica

- Sequência Lógica
- Instruções
- Algoritmo
- Programa
- Software



Lógica

A **lógica** de programação é a técnica de encadear pensamentos para atingir determinado objetivo.

Estes pensamentos podem ser descritos como uma sequência de instruções que devem ser seguidas para cumprir uma determinada tarefa



Sequência Lógica

Sequência lógica são os passos executados até atingir um
objectivo ou solução de um problema



Instruções

Uma **instrução** é uma informação que indica ao computador uma acção elementar a executar



Algoritmo

Sequência finita de passos que levam
à execução de uma tarefa.

Uma sequência de instruções que permitem
atingir um objectivo específico



Programa

Algoritmo escrito numa linguagem de programação

Software

Conjunto de programas

Algoritmos

Algoritmo

Um **algoritmo** é uma sequência de instruções delimitadas, um conjunto de passos claros e objectivos que permitem resolver um problema ou problemas.

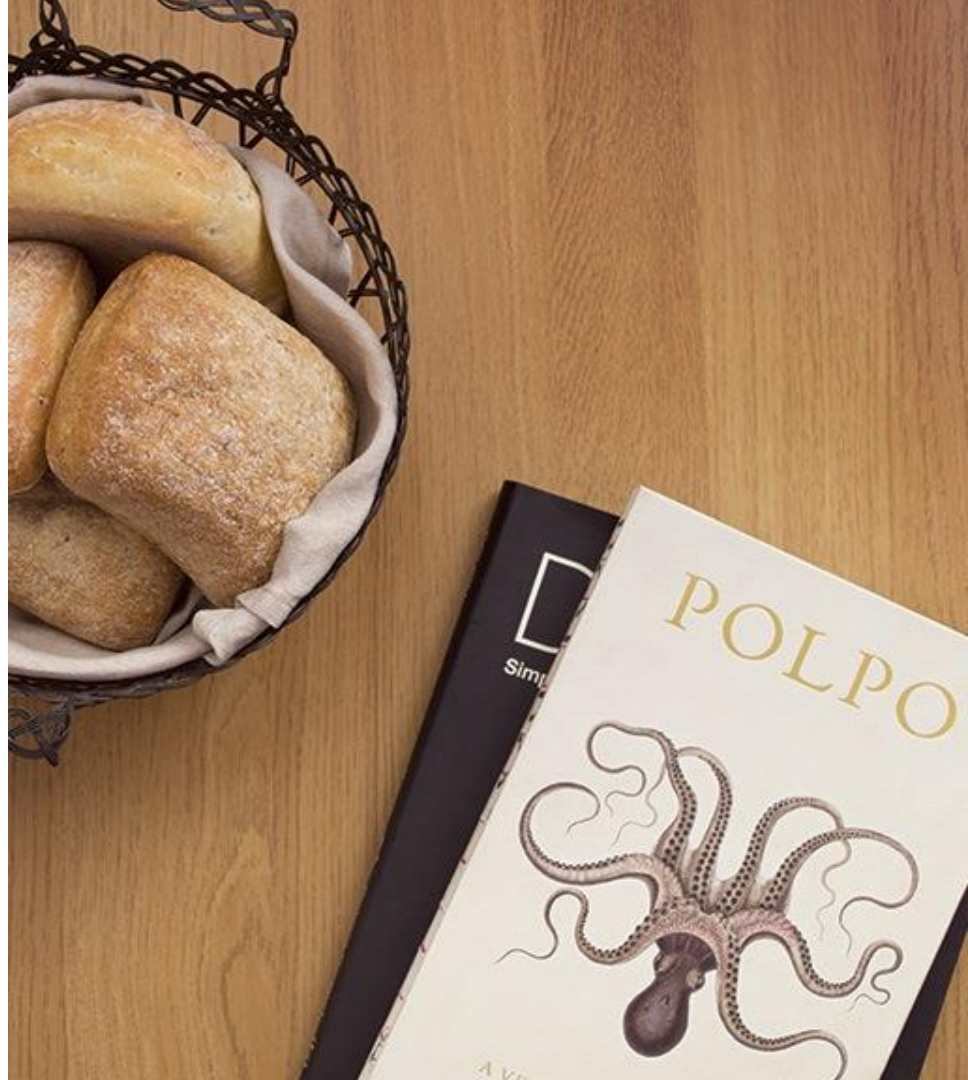
O acto de construir algoritmos é **programar**

Ao contrário de uma linguagem de programação, não existe um formalismo rígido para a escrita de um algoritmo.



Algoritmo

Como:
Livro de receitas, Manual de
instruções, ...



5 Características de um algoritmo

Finitude

Definitude

Entrada

Saída

Eficácia



5 Características de um algoritmo

Finitude

Todos os passos definidos num algoritmo têm de ter um fim previsto, não podem ser deixados condicionantes que levem à infinitude

Definitude

Entrada

Saída

Eficácia



5 Características de um algoritmo

Finitude

Definitude

Um algoritmo e os seus passos têm de ser definidos até à exaustão, não pode ser deixada qualquer ambiguidade que leve à indefinição de instruções.

Entrada

Saída

Eficácia



5 Características de um algoritmo

Finitude

Definitude

Entrada

Para a definição de um algoritmo é necessário identificar os dados de entrada, se existirem.

Saída

Eficácia



5 Características de um algoritmo

Finitude

Definitude

Entrada

Saída

O algoritmo procura a resolução de um problema; deverá produzir dados de saída, sendo necessário identificar os dados ou métodos de saída.

Eficácia



5 Características de um algoritmo

Finitude

Definitude

Entrada

Saída

Eficácia

Os passos de um algoritmo não devem ultrapassar um nível de complexidade que não possam ser entendidos ou executados pelo programador evitando problemas a implementar, testar e corrigir.



Fases de um Algoritmo

- ENTRADA
 - Dados de entrada do algoritmo;
- PROCESSAMENTO
 - Procedimentos utilizados para chegar ao resultado final;
- SAÍDA
 - Dados já processados.



Exercicio

- Pense numa tarefa diária, e cria uma sequência algoritmica, descritiva das várias fases dessa tarefa (dados de entrada, processamento, dados de saída)



Escrever Um Algoritmo

1

1

Definir o
Objectivo

O que
pretendemos
alcançar?

1

Definir o
Objectivo

O que
pretendemos
alcançar?

2

1

DEFINE THE
GOAL

What do you
want to
achieve?

2

Conceber o
Algoritmo

Dividir o
objectivo em
pequenas
instruções

1

Definir o
Objectivo

O que
pretendemos
alcançar?

2

Conceber o
Algoritmo

Dividir o
objectivo em
pequenas
instruções

3

1

Definir o
Objectivo

O que
pretendemos
alcançar?

2

Conceber o
Algoritmo

Dividir o
objectivo em
pequenas
instruções

3

“Codificar”
cada
instrução

Utilizando
uma
linguagem de
programação

Descrição de um Algoritmo (2)

- Pseudocódigo
- Fluxogramas ou Diagramas de Fluxos de Dados (DFD)

Pseudocódigo

Pseudocódigo é uma alusão à fase anterior à implementação do código numa linguagem de programação.

Um algoritmo em **pseudocódigo** é um intermediário entre a linguagem falada e a linguagem de programação.



Exemplo 1

INICIO

Colocar a roupa na máquina de lavar

Iniciar o programa de lavagem

Retirar a roupa da máquina

FIM



Exemplo 2

inicio

logico aberto

escrever "Porta aberta? "

ler aberto

se aberto **entao**

escrever "Sim"

senao

escrever "Não"

fimse

fim

Fluxogramas

Representação padronizada do **algoritmo em diagramas**.

Implica a utilização de **símbolos definidos na descrição da sequência de instruções e processos lógicos** de um algoritmo.

É mais objetivo e sintético que o pseudocódigo.

Exige o conhecimento dos **símbolos e métodos de representação** de processos.



CUSTOM SIGNAGE

Enter name:

SHOW COST

CUSTOM SIGNAGE

Enter name: Please enter a name below...

THOMAS

SHOW COST

CUSTOM SIGNAGE

\$30

T H O M A S



CUSTOM SIGNAGE

Enter name:

SHOW COST

CUSTOM SIGNAGE

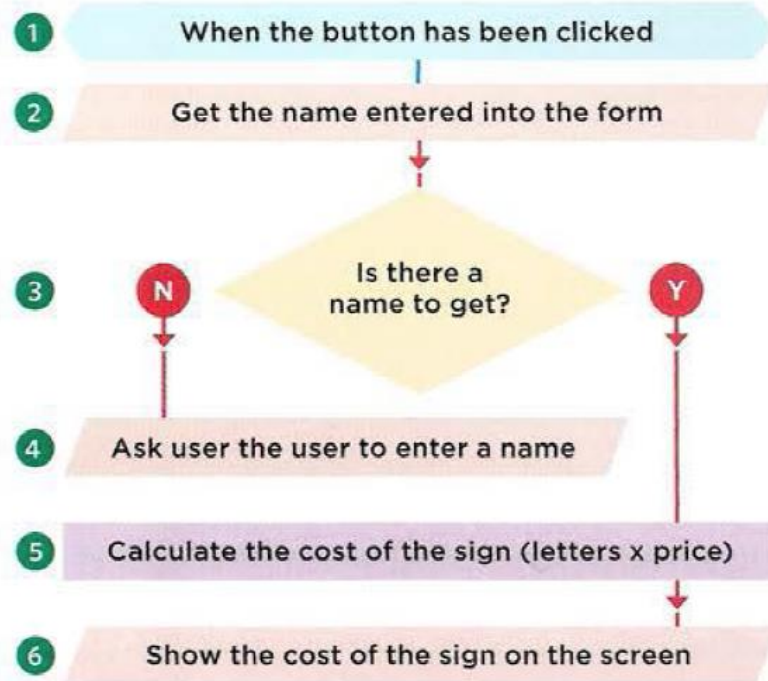
Enter name: Please enter a name below...

SHOW COST

CUSTOM SIGNAGE

THOMAS

\$30



Constantes, Variáveis e Tipos de Dados

Constantes, Variáveis e Tipos de Dados

- As **variáveis** e as **constantes** são os dados que um programa manipula
- São espaço reservado na memória do computador para armazenar um determinado **tipo de dados**
- Um programa deve conter declarações que **especifiquem de que tipo são os dados**; por vezes indicam o seu valor inicial





Value of the constant is not changed during execution of the program

Usage: use to declare something that won't be changed during program execution

Constante

- Valor fixo que não se modifica ao longo do tempo de execução de um programa

$$N1 + N2 + N3$$

3



Value of the variable can be changed anytime during execution of the program

Usage: use to store data that might change during program execution

Variável

- Cada variável corresponde a uma posição de memória, cujo valor pode ser alterado ao longo do tempo durante a execução de um programa
- Embora uma variável possa assumir diferentes valores, só pode armazenar um valor a cada instante

$N1 + N2 + N3$

3

Tipos de dados

- **Numéricas (Numbers)**
Específicas para armazenamento de números, que posteriormente poderão ser utilizados para cálculos;
- **Caracteres (Strings)**
Específicas para dados que contenham letras e/ou números;
- **Lógicas (Booleanos)**
Armazenam apenas dados lógicos que podem ser Verdadeiro ou Falso

Operadores

Operadores

Elementos que permitem
incrementar, decrementar,
comparar e avaliar dados



Tipos de Operadores

- Aritméticos
- Relacionais
- Lógicos



Operadores Aritméticos

Usados para obter resultados aritméticos

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
**	Exponenciação
%	Resto da divisão inteira



Operadores Relacionais

Usados para comparar dados, retornam verdadeiro ou falso

==	Igual
<>, !=	Diferente
>	Maior que
<	Menor que
>=	Maior ou igual que
<=	Menor ou igual que



Operadores Unários

Usados para atualizar a própria variável

++x	Pré-incremento
--x	Pré-decremento
x++	Pós-incremento
x--	Pós-decremento
!x	Negação



Operadores Lógicos

Usados para combinar operadores relacionais,
retornam verdadeiro ou falso

E / AND / &&	Todas as condições se verificam
OU / OR /	Pelo menos uma das condições se verifica
NÃO / NOT / !	Inverte o resultado da expressão ou condição



Estruturas de Decisão e Repetição

Estruturas de Decisão

É comum ser necessário tomar decisões de acordo com os dados avaliados, ou seja, executar blocos diferentes de instruções de acordo com esses dados.

Se... Então	Executar se a condição se verifica
Se... Então... Senão	Executar se a condição se verifica, caso contrário executar outro bloco
Caso... Seleccione	Se o valor da variável critério é igual a uma das N possibilidades dadas, é executado o bloco de instruções correspondente.

Estruturas de Repetição

Utilizam-se para executar um conjunto de instruções enquanto uma expressão se verifica ou até que se verifique

Enquanto ... Fazer	Do While... Loop
Até que ... Fazer	Do Until... Loop
Fazer... Enquanto ...	Do... Loop While
Fazer... Até que ...	Do... Loop Until
De... Até... Seguinte	For... To... Next

Estudo de Caso #1

LightBot

Solve Puzzles using Programming Logic.

<http://lightbot.com>



Pseudocódigo

Pseudocódigo

Pseudocódigo é uma alusão à fase anterior à implementação do código numa linguagem de programação.

Um algoritmo em **pseudocódigo** é um intermediário entre a linguagem falada e a linguagem de programação.



Fluxogramas

ou Diagramas de Fluxos de Dados (DFD)

Fluxogramas

Representação padronizada do **algoritmo em diagramas**.

Implica a **utilização de símbolos definidos na descrição da sequência de instruções e processos lógicos** de um algoritmo.

É **mais objetivo** e sintético que o pseudocódigo.

Exige o conhecimento dos símbolos e métodos de representação de processos.



Símbolos



Processo



processo
pré-definido



decisão



decisão múltipla



Output
saída de dados (doc.)



entrada/saída
dados



terminador
(início e fim)



conector
(pode ser numerado quando n)



entrada manual
(ex. teclas, texto)



operação manual
(entrada de dados)



display
(output para ecrã)



fluxo
(ligação entre símbolos)

Estudo de Caso #2

PORTUGOL

Simulador de linguagem algorítmica



PortugolIDE

<http://orion.ipt.pt/~manso/Portugol/download.html>

Exemplos 1 (decisao)

Se ... Entao ... Senao

Executa uma instrução caso uma condicao seja verdadeira, caso contrário executa outra

inicio

inteiro num

ler num

se num \geq 10 **entao**

escrever "O numero ", num, " e maior ou igual que 10!"

senao

escrever "O numero ", num, " e menor que 10!"

fimSe

fim

Exemplos (decisão)

CASO

Esta estrutura compara a expressão_0 com as restantes expressões (expressão_1 . . . expressão) e se a comparação for positiva executa as instruções do Caso e sai.

A estrutura tem um caso por defeito (opcional) que é executado quando todas as outras condições falharem.

inicio

caracter genero

escrever "Qual o seu genero?"

ler genero

escrever "Olá "

escolhe genero

caso "m" :

escrever "Homem"

caso "f" :

escrever "Mulher"

defeito :

escrever "genero indefinido"

fimescolhe

escrever ", bem vindo "

fim

Exemplos 1 (repeticao)

Enquanto

Repete as instruções enquanto a condição for verdadeira

inicio

inteiro num <- 0

enquanto num < 10 **faz**

escrever num, "\t"

num <- num + 1

fimEnquanto

fim

Exemplos 2 (repeticao)

Repete

Repete as instruções até que a condição seja verdadeira.

inicio

inteiro num <- 10

repete

escrever num, " - "

num <- num -1

ate num < 0

fim

Exemplos 3 (repeticao)

Para

Repete as instruções controlado por uma variável numérica que percorre os valores entre dois limites utilizando o passo definido.

Caso o passo não seja definido, o passo é de mais ou menos uma unidade de forma a poder percorrer o intervalo entre o valor inicial e o final de forma crescente ou decrescente.

inicio

inteiro num

para num **de** 1 **ate** 10 **passo** 1
escrever num, " - "

proximo

fim

Exemplos 4 (repeticao)

FAZ

Repete as instruções até
que a condição seja falsa.

inicio

inteiro mes

faz

escrever "introduza um mes"

ler mes

enquanto mes < 0 ou mes > 13

escrever "\nmes introduzido: ", mes

fim

Estudo de Caso #2.1

inicio

logico aberto

escrever "Porta aberta? "

ler aberto

se aberto **entao**

escrever "Sim"

senao

escrever "Não"

fimse

fim



Estudo de Caso #2.2

Usando o PORTUGOL escreva um programa que represente o comportamento do termostato de um Ar Condicionado.

Se a temperatura for maior ou igual a 23 graus ligar no frio, caso contrário ligar no quente.

Estudo de Caso #2.3

Usando o PORTUGOL escreva um programa que peça um inteiro e indique se o inteiro indicado é par ou ímpar

Estudo de Caso #2.4

Usando o PORTUGOL escreva um programa que calcule o volume de um objecto

Nota: profundidade X altura X largura



Estudo de Caso #2.5

Usando o PORTUGOL escreva um programa que solicite o nome e o género do utilizador. Com esses dados deve imprimir: “Olá Sr. X” ou “Olá Sra. X”



Estudo de Caso #2.6

Usando o PORTUGOL escreva um programa que solicite o número de alunos e calcule a média de notas de uma turma



Estudo de Caso #2.7

Usando o PORTUGOL escreva um programa que solicite 2 notas e indique qual a media:

Mediante a media indique:

- se chumbou < 4 valores

- se passou ≥ 7

- se repete exame ≥ 4 e < 7

